PENJELASAN DARI KODE PEMBELAJARAN

PERTEMUAN 4

Nama : Muuhammad Luqmqnul Fikri

NIM : 231011400546

Kelas : 05TPLE017

□ 1. Import Library Utama

```
# --- Import library utama ---
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
```

📓 Penjelasan:

- `pandas (pd)` → untuk membaca, menampilkan, dan mengolah data dalam bentuk tabel (DataFrame).
- `matplotlib.pyplot (plt)` → membuat grafik dasar (plot, histogram, scatter).
- `seaborn (sns)` → membuat visualisasi data yang lebih menarik dan mudah dibaca.
- `train_test_split` → fungsi dari `scikit-learn` untuk membagi dataset menjadi **data latih**, **validasi**, dan **uji**.

📂 2. Load Dataset

```
# --- 1. Load dataset ---
df = pd.read_csv("processed_kelulusan.csv")  # pastikan nama file sesuai
```

📓 Penjelasan:

- Membaca file CSV bernama `processed_kelulusan.csv` dan menyimpannya ke variabel `df`.
- `df` sekarang berisi seluruh data yang akan digunakan dalam pemodelan machine learning.

🧱 3. Membuat Fitur Baru (Feature Engineering)

```
# --- 2. Bikin fitur baru ---
df["Rasio_Absensi"] = df["Jumlah_Absensi"] / 14
df["IPK_x_Study"] = df["IPK"] * df["Waktu_Belajar_Jam"]
```

📓 Penjelasan:

- **Tujuan:** menambah kolom baru yang bisa memperkuat model (fitur turunan).
- `Rasio_Absensi` → persentase kehadiran, dengan mengasumsikan total pertemuan = 14 kali.
- `IPK_x_Study` → hasil kali antara IPK dan jam belajar per minggu (indikator kombinasi prestasi dan usaha belajar).

🔍 4. Mengecek Struktur Data

```
# --- 3. Cek data ---
print(df.info())
print(df.head())
```

📓 Penjelasan:

- `df.info()` → menampilkan tipe data setiap kolom, jumlah nilai kosong (null), dan total data.
- `df.head()` → menampilkan 5 baris pertama dataset untuk memastikan data sudah benar terbaca.

📊 5. Statistik Deskriptif

```
# --- 4. Statistik deskriptif ---
print("\nStatistik deskriptif:")
print(df.describe())
```

📓 Penjelasan:

- `df.describe()` → menampilkan ringkasan statistik seperti mean, min, max, std, dan quartiles untuk kolom numerik.
- Berguna untuk melihat **sebaran data** dan mendeteksi **nilai ekstrem/outlier**.

☑ 6. Visualisasi (Exploratory Data Analysis / EDA)

Bagian ini menampilkan **pola, hubungan, dan distribusi data** menggunakan grafik.

◆ Korelasi antar fitur

```
# --- 5. Visualisasi (EDA) ---
plt.figure(figsize=(6,5))
sns.heatmap(df.corr(), annot=True, cmap="coolwarm", center=0)
plt.title("Correlation Heatmap")
plt.show()
```

- `df.corr()` → menghitung korelasi antar variabel numerik.
- `sns.heatmap` → menggambar peta warna hubungan antar variabel (positif = merah, negatif = biru).
- Berguna untuk melihat fitur mana yang berhubungan kuat dengan target `Lulus`.

◆ Distribusi kelas target

```
# distribusi kelas target
plt.figure(figsize=(4,3))
sns.countplot(x="Lulus", data=df)
plt.title("Distribusi Kelas Target")
plt.show()
```

- Menampilkan jumlah mahasiswa yang **lulus (1)** dan **tidak lulus (0)**.
- Membantu memastikan data target **seimbang** (tidak terlalu banyak 1 dibanding 0).

◆ Hubungan antara IPK dan Waktu Belajar

```
# scatterplot hubungan IPK - Jam Belajar
plt.figure(figsize=(5,4))
sns.scatterplot(x="IPK", y="Waktu_Belajar_Jam", hue="Lulus", data=df, palette="Set1")
plt.title("IPK vs Jam Belajar (dengan Label Lulus)")
plt.show()
```

- Scatter plot untuk melihat apakah **mahasiswa dengan IPK tinggi dan jam belajar lebih banyak** cenderung **lebih sering lulus**.

◆ Distribusi IPK

```
# histogram IPK
plt.figure(figsize=(5,3))
sns.histplot(df["IPK"], bins=10, kde=True)
plt.title("Distribusi IPK")
plt.show()
```

- Menunjukkan sebaran nilai IPK seluruh mahasiswa.
- `kde=True` menambahkan kurva distribusi halus di atas histogram.

🔧 7. Split Dataset (Train, Validation, Test)

```
# --- 6. Split dataset: Train, Validation, Test ---
X = df.drop("Lulus", axis=1)
y = df["Lulus"]
```

- `X` = semua kolom fitur (tanpa kolom target `Lulus`)
- `y` = kolom target (`Lulus`, biasanya 0 atau 1)

📓 Langkah 1 – Membagi Train dan Temp

```
# pertama bagi Train + Temp
X_train, X_temp, y_train, y_temp = train_test_split(
    X, y, test_size=0.3, stratify=y, random_state=42
)
```

- 70% data digunakan untuk **pelatihan (train)**.
- 30% sisanya disimpan sementara di `X_temp` untuk nanti dibagi lagi jadi **validasi dan test**.
- `stratify=y` menjaga agar proporsi kelas target tetap seimbang.

📓 Langkah 2 – Membagi Temp jadi Validation dan Test

```
# lalu bagi Temp jadi Validation + Test
X_val, X_test, y_val, y_test = train_test_split(
    X_temp, y_temp, test_size=0.5, stratify=y_temp, random_state=42
)
```

- Membagi data `X_temp` menjadi dua:
  - 50% untuk **validasi (val)**
  - 50% untuk **uji akhir (test)**
- Hasil akhirnya:
  - Train = 70%
  - Validation = 15%
  - Test = 15%

🔹 Menampilkan ukuran masing-masing set:

```
print("\nUkuran dataset:")
print("Train:", X_train.shape, " Validation:", X_val.shape, " Test:", X_test.shape)
```

Menampilkan jumlah baris dan kolom setiap subset data.

💾 8. Simpan Dataset Bersih

```
# --- 7. Simpan dataset bersih ---
df.to_csv("processed_kelulusan.csv", index=False)
print("\nFile processed_kelulusan.csv berhasil dibuat!")
```

- Menyimpan kembali dataset yang sudah diproses dan ditambah fitur baru.
- Berguna untuk digunakan di tahap **pelatihan model machine learning** berikutnya.

✳ Kesimpulan:

Kode ini melakukan:

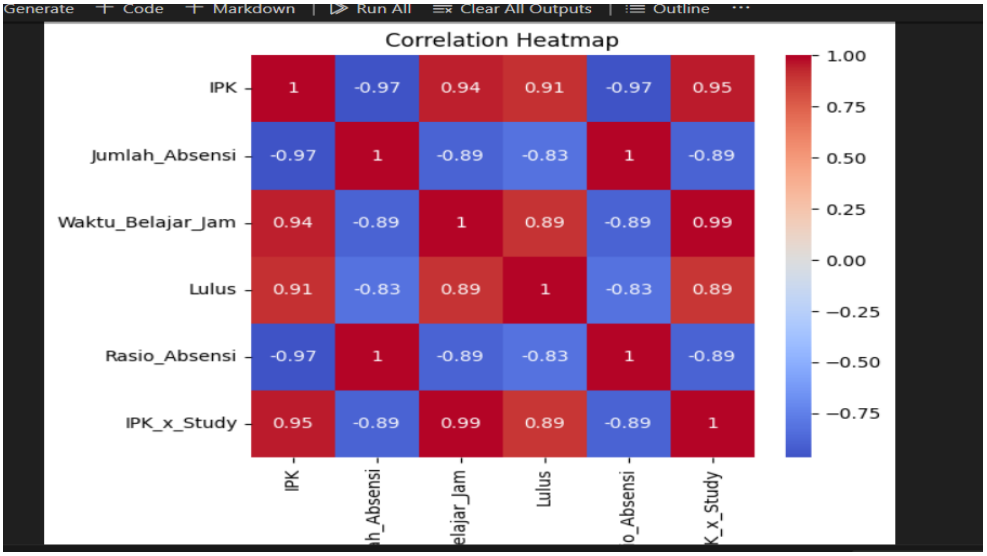
1. Membaca dataset dan menambah fitur baru.
2. Menampilkan struktur dan statistik data.
3. Melakukan visualisasi (EDA) untuk memahami hubungan antar variabel.
4. Membagi data menjadi **train–validation–test** secara proporsional.
5. Menyimpan dataset bersih siap digunakan untuk pemodelan **Machine Learning** berikutnya

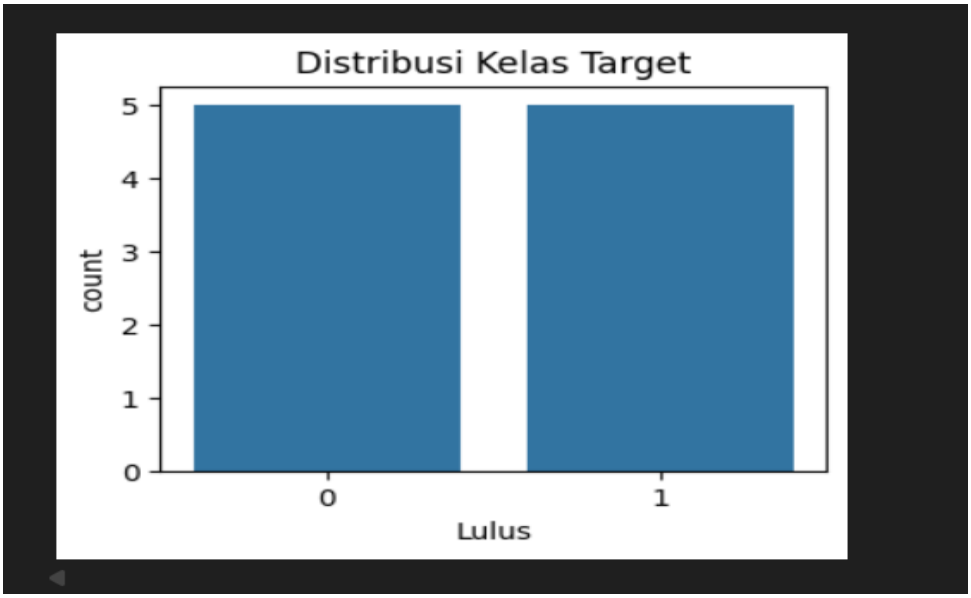
HASILNYA

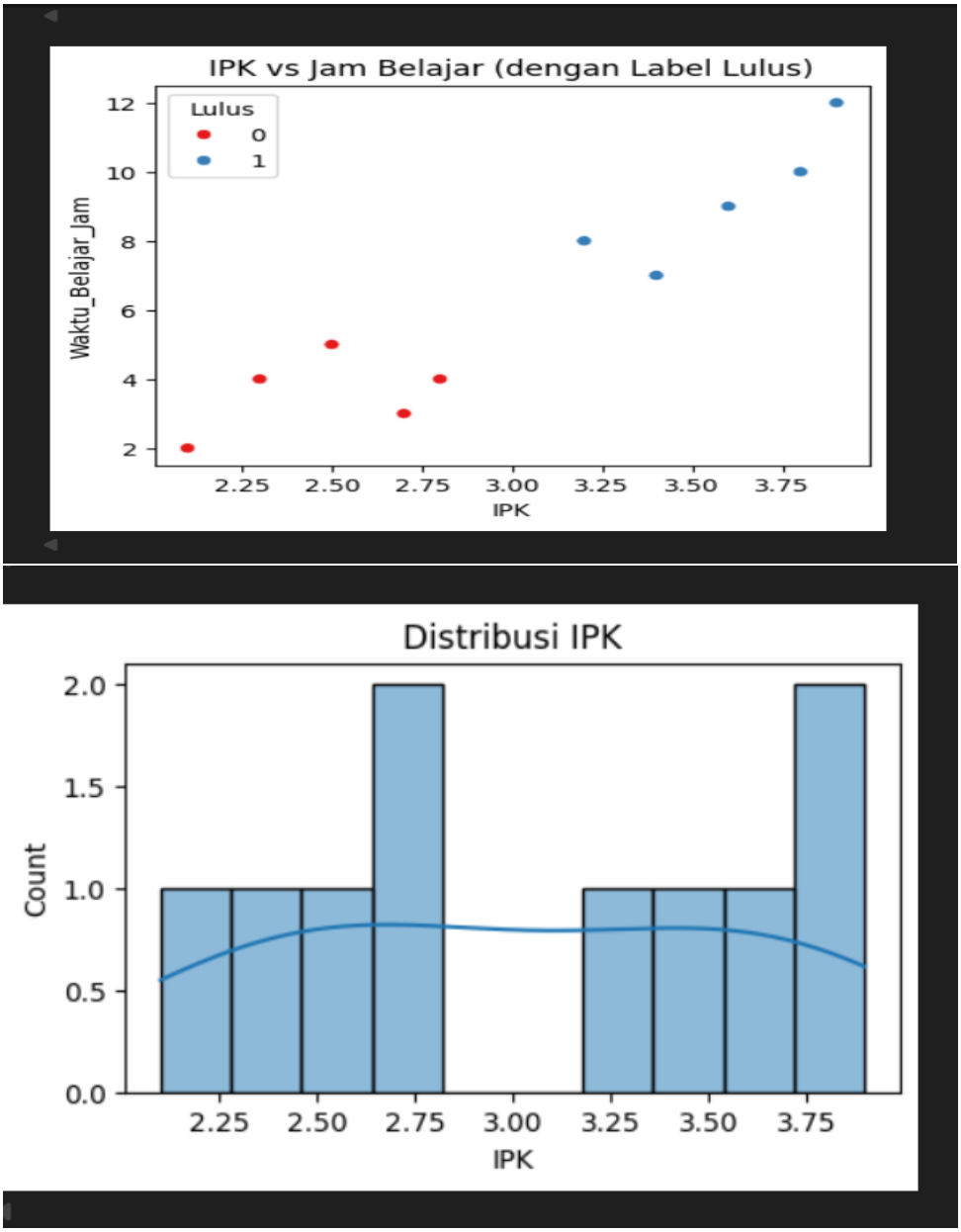
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 6 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   IPK               10 non-null     float64
 1   Jumlah_Absensi    10 non-null     int64  
 2   Waktu_Belajar_Jam  10 non-null     int64  
 3   Lulus             10 non-null     int64  
 4   Rasio_Absensi     10 non-null     float64
 5   IPK_x_Study       10 non-null     float64
dtypes: float64(3), int64(3)
memory usage: 608.0 bytes
None
   IPK  Jumlah_Absensi  Waktu_Belajar_Jam  Lulus  Rasio_Absensi  IPK_x_Study
0  3.8               3                 10      1       0.214286         38.0
1  2.5               8                  5      0       0.571429         12.5
2  3.4               4                  7      1       0.285714         23.8
3  2.1              12                  2      0       0.857143          4.2
4  3.9               2                 12      1       0.142857         46.8

Statistik deskriptif:
             IPK  Jumlah_Absensi  Waktu_Belajar_Jam      Lulus  Rasio_Absensi  \
count  10.000000        10.00000          10.000000  10.000000      10.000000
mean    3.030000         6.00000           6.400000   0.500000       0.428571
```

```
...
25%       9.700000
50%      18.150000
75%      30.700000
max      46.800000
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```


Generate
Code
Markdown
Run All
Clear All Outputs
Outline
Correlation Heatmap
IPK
Jumlah_Absensi
Waktu_Belajar_Jam
Lulus
Rasio_Absensi
IPK_x_Study
1 -0.97 0.94 0.91 -0.97 0.95
-0.97 1 -0.89 -0.83 1 -0.89
0.94 -0.89 1 0.89 -0.89 0.99
0.91 -0.83 0.89 1 -0.83 0.89
-0.97 1 -0.89 -0.83 1 -0.89
0.95 -0.89 0.99 0.89 -0.89 1
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
IPK
Lulus


Distribusi Kelas Target
count
0
1
2
3
4
5
0
1
Lulus


IPK vs Jam Belajar (dengan Label Lulus)
Lulus
0
1
Waktu_Belajar_Jam
12
10
8
6
4
2
2.25
2.50
2.75
3.00
3.25
3.50
3.75
IPK

Distribusi IPK
Count
2.0
1.5
1.0
0.5
0.0
2.25
2.50
2.75
3.00
3.25
3.50
3.75
IPK